Radiative Signatures of the Relativistic Kelvin-Helmholtz Instability

M. Bussmann
Helmholtz-Zentrum Dresden – Rossendorf
Bautzner Landstrasse 400
01328 Dresden, Germany
+49 351 260-2616
m.bussmann@hzdr.de

H. Burau
Helmholtz-Zentrum Dresden – Rossendorf
Bautzner Landstrasse 400
01328 Dresden, Germany
+49 351 260-3377
h.burau@hzdr.de

T.E. Cowan
Helmholtz-Zentrum Dresden – Rossendorf
Bautzner Landstrasse 400
01328 Dresden, Germany
+49 351 260-2270
t.cowan@hzdr.de

A. Debus
Helmholtz-Zentrum Dresden – Rossendorf
Bautzner Landstrasse 400
01328 Dresden, Germany
+49 351 260-2619
a.debus@hzdr.de

A. Huebl
Helmholtz-Zentrum Dresden – Rossendorf
Bautzner Landstrasse 400
01328 Dresden, Germany
+49 351 260-3639
a.huebl@hzdr.de

G. Juckeland
Technische Universität Dresden
Center for Information Services and High Performance Computing
01062 Dresden, Germany
+49 351 463-39871
guido.juckeland@tu-dresden.de

T. Kluge
Helmholtz-Zentrum Dresden – Rossendorf
Bautzner Landstrasse 400
01328 Dresden, Germany
+49 351 260-2618
t.kluge@hzdr.de

W.E. Nagel
Technische Universität Dresden
Center for Information Services and High Performance Computing
01062 Dresden, Germany
+49 351 463-35450
wolfgang.nagel@tu-dresden.de

R. Pausch
Helmholtz-Zentrum Dresden – Rossendorf
Bautzner Landstrasse 400
01328 Dresden, Germany
+49 351 260-2616
r.pausch@hzdr.de

F. Schmitt
Technische Universität Dresden
Center for Information Services and High Performance Computing
01062 Dresden, Germany
+49 351 463-39871
felix.schmitt@tu-dresden.de

U. Schramm
Helmholtz-Zentrum Dresden – Rossendorf
Bautzner Landstrasse 400
01328 Dresden, Germany
+49 351 260-2618
u.schramm@hzdr.de

J. Schuchart
Oak Ridge National Lab
PO Box 2008 MS-6164
Oak Ridge, TN 37831-6164
+1 865 241-6293
schuchartj@ornl.gov

R. Widera
Helmholtz-Zentrum Dresden – Rossendorf
Bautzner Landstrasse 400
01328 Dresden, Germany
+49 351 260-3543
r.widera@hzdr.de

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.

Copyright is held by the owner/author(s).

/SC '13/, Nov 17-21 2013, , USA
ACM 978-1-4503-2378-9/13/11.
http://dx.doi.org/10.1145/2503210.2504564

ABSTRACT

We present a particle-in-cell simulation of the relativistic Kelvin-Helmholtz Instability (KHI) that for the first time delivers angularly resolved radiation spectra of the particle dynamics during the formation of the KHI. This enables studying the formation of the KHI with unprecedented spatial, angular and spectral resolution. Our results are of great importance for understanding astrophysical jet formation and comparable plasma phenomena by relating the particle motion observed in the KHI to its radiation signature.

The innovative methods presented here on the implementation of the particle-in-cell algorithm on graphic processing units can be directly adapted to any many-core parallelization of the particle-mesh method. With these methods we see a peak performance of 7.176 PFLOP/s (double-precision) plus 1.449 PFLOP/s (single-precision), an efficiency of 96% when weakly scaling from 1 to 18432 nodes, an efficiency of 68.92% and a speed up of 794 (ideal: 1152) when strongly scaling from 16 to 18432 nodes.

1. SUBMISSION CATEGORIES

- Scalability
- Peak Performance
- Time-to-Solution

2. OVERVIEW OF THE PROBLEM AND ITS IMPORTANCE

The Kelvin-Helmholtz instability (KHI) is one of the fundamental instabilities observed in plasmas. Originally studied in fluids [62, 61], the plasma KHI nowadays plays an important role in such diverse fields as Astrophysics [22, 57, 54], Space Physics [35, 39], Solar Physics [17, 40, 19], Planetary Physics [51, 13], Fusion Research [9, 53], and Laser-Plasma Physics [30, 34, 55].

In this work we study the KHI between two relativistic, unmagnetized streams of hydrogen plasma. The KHI occurs when two streams of plasma flow past each other. Protons and electrons in the stream flow at a velocity corresponding to a relativistic γ-factor of 3. At the interface between the streams, shear forces break up the flow and the plasma flow becomes turbulent, forming current filaments. These filaments amplify a magnetic field that again leads to a stronger perturbation of the shear surface creating a feedback loop. During this phase, certain electromagnetic wave modes grow and periodic magnetic field structures appear, interacting with the density modulations. Finally, a quasi-steady state is reached which manifests in a periodic density and field modulation along the interface between the streams.

The onset and development of the instability is of great interest in astrophysical scenarios where a relativistic particle jet passes through background plasma. Kinetic simulations using the particle-in-cell (PIC) technique give insight into the interplay between particle kinetics responsible for the density perturbations and the formation of the fields. Only recently, large-scale particle-in-cell simulations revealed that transverse KHI dynamics and, for the relativistic case, strong DC fields are observed [3] that have not been seen in hydrodynamic KHI simulations.

The dynamics of the relativistic KHI in astrophysical scenarios cannot be observed directly, however, the radiation emitted by the electrons during the temporal evolution of the KHI can serve as a signature of the plasma dynamics [42]. Up until now, kinetic simulations of the relativistic KHI did not include the computation of the far field radiation emission, thus missing an essential diagnostic for understanding astrophysical particle acceleration.

For the first time a 3D fully-relativistic particle-in-cell simulation of the KHI is attempted which includes the far-field radiation of the electrons. The radiation emitted by the electrons is computed for the full wavelength range of interest which spans over three orders of magnitude. Radiation spectra are recorded for 481 directions, each including 512 frequencies in a range between 1.4% of the plasma frequency to 14 times the plasma frequency ω_p. One quadrant of the full solid angle is covered as the region of emission, making use of the intrinsic symmetry of the system.

Our calculation was performed in the rest frame of the streams, which enables us to compute the KHI dynamics in a large volume that contains 9 vortices excited by the KHI. It furthermore reduces the demands on the field solver. At high relativistic gamma factors, denoting high jet energies, field solvers in particle-in-cell codes suffer from so-called numerical Cherenkov radiation that can only be suppressed by heavy filtering of electromagnetic waves or by reducing the accuracy of the charge conservation method [23]. Such schemes may influence the particle wave interaction. We compute the spectrally-resolved radiation power in one quadrant of the full solid angle and thus keep the system size large in order to minimize statistical noise in the spectra. Our results can in principle be transformed into Lorentz-boosted frames of reference to compare to the emission of high energy jets [42].

The simulation volume consists of 8000 x 768 x 768 cells with 8 protons and 8 electrons per cell with a mass ratio of 1836:1 and a charge ratio of 1:1. This results in a total number of 7.5×10^{10} particles simulated, while half of all electrons are used for calculating the far field radiation. This gives an unprecedented spatial resolution of 0.06 classical skin depths (ω_p/c) with a total volume of 480 x 46 x 46 classical skin depths computed, thus including a total of 9 KHI wavelengths. The total simulation time covers $62\omega_p^{-1}$ or a total of 2000 time steps. We chose the field solver of Yee [67], the Boris particle push [5] and the Esirkepov current deposition method [16] with a TSC macro-particle shape function [28] for all simulations. Such a large-scale particle-in-cell simulation of the Kelvin-Helmholtz instability with angular resolved spectra has to our best knowledge never been attempted before and allows for the first time to connect the microscopic electron dynamics to the radiated spectrum with unprecedented angular and spectral resolution. With our results it will become possible to understand in detail the electron dynamics in relativistic counter-streaming plasmas by observing their spectral signature, opening new paths in observation-based astrophysics.

3. CURRENT STATE OF THE ART

The Particle-in-Cell (PIC) algorithm [12] computes the interaction between charged macro-particles by solving the field equations on a mesh. Unlike hydrodynamic plasma codes, PIC codes do not assume a certain plasma equation of state. Local plasma density, temperature and pressure changes can be resolved by the macro-particle densities, kinetic energies and momenta. Turbulent and nonlinear wave phenomena can be much better resolved than in magneto-hydrodynamic (MHD) plasma simulations, albeit at a much greater computational cost, as the plasma phase space has to be sampled by a large amount of macro-particles. Thus, up until very recently, large-scale PIC simulations of the KHI have not been feasible at all [3, 36, 42].

To our knowledge, the largest KHI simulation yet reported simulated a volume of 1000 x 320 x 320 cells with a spatial resolution of 0.25 classical skin depths [3]. In comparison, the cell volume simulated in this work is 46 times larger and the spatial resolution is 4.2 times higher. This makes it possible to observe the collective emission of 9 longitudinal vortices formed in the KHI with unprecedented resolution. For certain angles of observation, the radiation emitted by the vortices generates a combined signal, which requires a large number of particles and vortices to be included in the simulation in order to correctly estimate the spectral power of the signal.

The PIC algorithm is hard to parallelize on many-core systems mainly due to two reasons. First, data locality requires resorting of particle and field data, as particles propagate between the mesh cells over the course of the simulation. Thus, domain decomposition of both particle and field data results in particle data becoming nonlocal over time. Second, the computation of the current from the individual charged particle motion is a local reduction operation and thus always has a serial part. This so-called *current deposition* is one of the most computational intense algorithmic steps of the particle-in-cell algorithm.

For these reasons, particle-in-cell codes that make efficient use of many-core architectures are sparse. Moreover, the intrinsic serial nature of the current deposition does normally not allow for good strong scaling. Nevertheless, some particle-in-cell codes have shown close-to Petaflop performance, such as the OSIRIS code on Sequoia [18], and the VPIC code on Roadrunner [6]. VPIC cited 0.374 Pflop/s sustained performance, while for OSIRIS 97% weak scaling efficiency and 75% strong scaling efficiency was reported using up to 256,000 cores on Sequoia, albeit no peak performance results have been published.

As of today, many-core parallelization has been integrated into PIC codes [60, 7, 66, 32, 37, 38, 10, 33, 4, 26] for both non-relativistic and relativistic plasma computations. Although Graphic Processing Units (GPUs) have gained ground in accelerated high performance computing, the problem of preserving data locality has prohibited the wide spread of PIC implementations on the GPU. Most GPU-enabled PIC codes implement some of the steps of the PIC algorithm, while very few implement the complete PIC algorithm on GPUs [29]. Published scaling and performance results for GPU-enabled PIC codes are thus almost non-existent. As an example, the code Picador showed a weak scaling efficiency of 64% on 512 GPUs while estimating a maximum 14% of the peak performance [4]. To our best knowledge, only PIConGPU [7, 29], the code used for this simulation, makes use of both data-parallelism and task-parallelism to saturate the load on the GPU, see the Innovations section for details.

As relativistic PIC codes are limited to computing the near-field emission of radiation inside the simulated volume [15], any information on the far-field is missing. Moreover, radiation is usually only accounted for up to the resolution of the mesh on which the electric and magnetic fields are discretized [14]. Only recently, the computation of far-field radiation has been included into relativistic PIC codes [59, 52, 26, 42]. The spectral power density emitted into each single direction is computed via Liénard-Wiechert potentials, incoherently adding all contributions of all particles. This operation is essentially a Fourier Transform over the macro-particle trajectory with respect to retarded time. Accurate spectra can only be obtained by using a large amount of particles and non-equidistant sampling of the trajectories, while random selection of macro-particles to compute spectra [43] does not give the desired resolution and can result in strong aliasing effects, distorting the spectra [52]. The largest number of macro-particles used to compute spectra has been 10^6 [20], 4 orders magnitude less than in this work. Offline calculation of radiation [11] using such a large number of macro-particles is simply prohibited by storage size and file system performance.

4. INNOVATIONS

PIConGPU [8] is a three-dimensional, fully relativistic electromagnetic particle-in-cell code. It is written in CUDA [45] C++ using meta-programming techniques and compile-time polymorphism [2] in order to abstract from hardware-dependent code while maintaining maximum performance. It can thus run in single-precision, double-precision and mixed-precision mode for optimizing floating point performance and adopt the code to a certain CUDA compute capability [48]. It uses parallel HDF5 [27] file output and online visualization.

4.1 Physics model

PIConGPU solves electric and magnetic fields on a regular rectangular Cartesian mesh following the Yee finite difference time domain method [67]. The equation of motion of the particles is solved in the so-called *particle push* step of the PIC algorithm. PIConGPU implements the methods proposed by Boris [5] and Vay [63]. The calculation of the current, the *current deposition* step, is fully charge-conserving and can be performed via the Villasenor-Buneman [64] scheme or the Esirkepov scheme [16]. Macro-particle shapes implemented include NGP, CIC, TSC and PCS [28]. From now on, macro-particles will often be simply called particles.

4.2 Vectorized cell and particle data

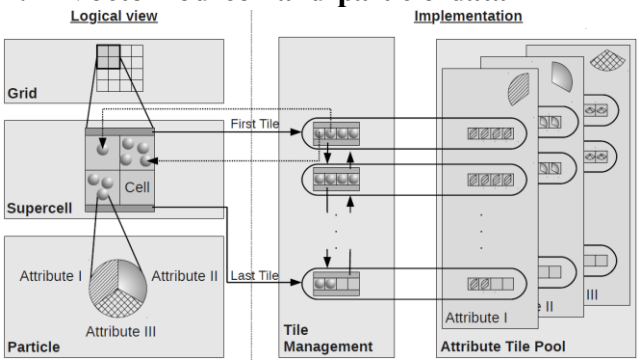

Figure 1. PIConGPU vectorized data formats for cell and particle data, super cell and attribute tile pool.

PIConGPU has introduced vectorized data structures for both cell-based field data and particle-based data. Meta-programming allows us to define multiple physical values discretized on staggered meshes, using a single abstract data type to handle electric field, magnetic field and current data. Similarly, particle data is represented by a single abstract data type to which particle attributes such as mass or charge state can be added, thus allowing for various particle species to be described efficiently.

Cell data is stored in a fixed-size data structure called *super cell*, a name introduced with PIConGPU now widely adopted by the community. It is located in shared memory and contains as many cells as there are threads executed in parallel in a thread block – in our simulation this amounts to 256 cells. When accessing cell data, each thread accesses the value in a single cell computed from its thread ID.

Particle data is stored in so-called *attribute tiles*. Unlike cell data, which should be accessible by more than one thread, particle data is usually accessed by a single thread and not shared between threads. Moreover, there are usually more particles than cells and fast memory access is vital. Thus, particle data is stored in register memory. Each attribute tile consists of the same number of elements as a super cell, namely the number of threads in a thread block. If there are more particles than cells in a super cell, additional attribute tiles are added. All attribute tiles are interconnected via a doubly-linked list and the first and last tile in the list are linked to the super cell. Each particle includes its current cell index in the super cell as an attribute. The attribute tile, like the super cell, is fully templated and particle attributes can be of any type, allowing for adopting the vectorized data types to the users needs.

Attribute tiles are fixed in size, which in the worst case leaves 99.6% of the memory allocated for the last tile unused if the particle number in a super cell is not an integer multiple of the number of threads in a thread block. This memory is used when particles enter the super cell. When a particle leaves the super cell, the entry in the tile is deleted, leaving a hole. For each tile holes are stored in a bitmap and are filled up on the end of each time step by reordering.

4.3 Data parallelism

With the super cell and attributes defined as above, easy thread-wise memory access to cell and particle data is ensured, as the thread ID can be directly associated to a single cell in a super cell or a single particle in an attribute tile, allowing for easy switching between cells and particles in a thread context. Global memory access is fully coalesced and optimized by aligning particle and cell data in global memory to 128 Byte size. Each thread block is assigned to a super cell and the corresponding attribute tile list, thus allowing for coalesced memory access. As long as particles stay inside the super cell, no memory has to be copied and all memory access is done via indexing. Thus, complete data parallel execution is achieved.

The most demanding algorithmic step in PIC is the current deposition. Depending on the macro-particle shape function, current data from a single particle has to be written to the cell in which the particle is centred and to its surrounding neighbour cells. As all particle-wise threads write current data in parallel to their according cell neighbourhood, write conflicts can occur. In PIConGPU, these conflicts are prevented by using atomic additions. As these act on the current data stored in shared memory only, the current deposition still performs well. Tests with an optimally sorted particle list and employing a parallel reduction over particles to compute the current did not yield much better performance than using atomic operations. Moreover, even parallel reduction becomes complex to implement if macro-particles deposit current to up to 64 cells (TSC macro-particle shape as used in the production run).

4.4 Asynchronous communication

The compute domain on each GPU is divided in a core and border region and is surrounded by ghost cells that provide memory for inter-GPU communication. This division of the compute domain allows for concurrent computation and communication. Kernels for cells in the border region are executed after those for the core, thus each step of the PIC algorithm can assume total data locality and memory transfer between GPU domains is hidden. Both host-device communication via cudaMemcpyAsync and inter-node communication via the Message Passing Interface (MPI) [41] Isend and Irecv are, thus, asynchronous. Individual ghost buffers for copying memory are introduced for each boundary to neighbouring GPUs and for each individual particle species and cell data. All data, especially higher-dimensional data, is copied to one-dimensional copy buffers as this yields better cudaMemcpyAsync performance. For each GPU a single thread on the CPU takes care of the communication and no computational tasks are carried out on the CPU. For parallel file output via HDF5 we invoke a second thread.

4.5 Task-parallelism

PIConGPU includes a complete event system that invokes parallel execution of kernels via CUDA streams. The PIC algorithm is divided into independent steps that can be executed in parallel [29]. Host-device communication between GPU and CPU is executed in parallel for all directions, individual particle attributes and individual cell data such as magnetic fields, electric fields and currents independently. Data dependencies are described by task graphs that define which kernels can run in parallel and which need input from other kernels. Task parallelism and thread parallelism together allow for an optimal load on the GPU.

As an example, consider the *current deposition* for a number of super cells performed by kernel A, containing more particles than the super cells handled by kernel B. The thread block for kernel A will usually take longer to execute than that for B. Kernels A and B can be executed in parallel. If B finishes earlier than A, it can update its *current* data for use by other kernels. This allows a potential third kernel C which requires *current* data from B but not from A to use this data for further calculations, resulting in optimum GPU load. The event system is defined in an abstract manner which allows us to replace cudaMemcpyAsync and asynchronous MPI calls by other means of communication.

4.6 Radiation calculation

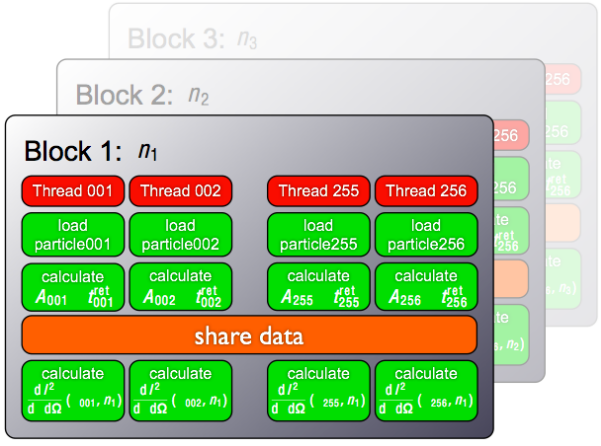

Figure 2. Parallelization scheme for computing the far field radiation via Liénard-Wiechert potentials.

Liénard-Wiechert potentials are used to compute the radiated energy per unit solid angle and unit frequency. The emitted spectral intensity $dI/d\omega$ of N particles into a solid angle $d\Omega$ can be calculated by a non-equidistant Fourier transform of the sum over the real amplitudes $\vec{A}_i$ over retarded time t_i^{ret} [31] for all particles.

$$\frac{d^2I}{d\omega d\Omega} = \frac{q^2}{16\pi^3\varepsilon_0 c}\left|\int_{t_{start}}^{t_{end}} \sum_{i=1}^{N} \vec{A}_i(t)\cdot e^{i\omega\cdot t_i^{ret}(t)} dt\right|$$

A discrete Fourier transform is used to allow for non-equidistant sampling of the trajectory and to minimize the memory used for storing trajectory information. Thread-wise parallelization over particles in a super cell is employed to calculate $\vec{A}_i$ and t_i^{ret} for each particle i. These results are stored in shared memory and used by each thread to compute the spectral intensity for a set of frequencies. Thread-block-wise parallelization is used to compute various observation angles. The total radiation is computed by incoherently summing up the results of all GPUs using MPI.

5. APPLICATIONS USED TO MEASURE PERFORMANCE

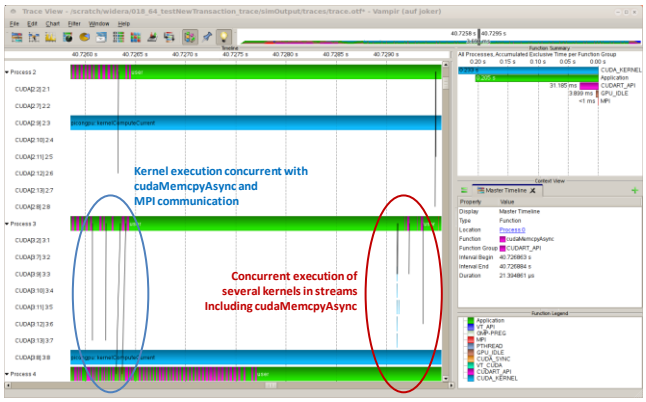

Figure 3. VAMPIR trace of PIConGPU kernel execution. Regions of concurrent host-hevice and inter-node communication (blue ellipse) and concurrent kernel execution (red ellipse) are highlighted.

PIConGPU has been developed by Helmholtz-Zentrum Dresden-Rossendorf in close collaboration with the VAMPIR performance optimization team of Technical University Dresden. PIConGPU performs all computational steps on the GPU, using CPU cores only for communication. Thus, traditional tools cannot be used to measure the performance of our code, as they lack support for GPU performance tracing. Performance measurements on TITAN [50] at Oak Ridge National Laboratory therefore made use of both VAMPIR TRACE 5.14.3 [21] and the NVIDIA command line profiler Nvprof [44] that is part of the CUDA 5.5 early access release [46]. Both VAMPIR TRACE and Nvprof use CUDA Profiling Tools Interface (CUPTI [47]) counters and metrics. We used the VAMPIR performance suite to determine the percentage of a single time step the simulation spends in executing computational kernels. This measurement was performed for 20 time steps and we state the average percentage in the performance results section.

With CUDA 5.5 two new metrics for measuring the number of floating point operations (FLOP) have been added to the CUDA Metric API, namely flops_sp for single-precision FLOP and flops_db for double-precision FLOP. Unfortunately, CUDA 5.5 early access was not available at TITAN at the time of this writing. For counting the number of floating point operations we thus used our local cluster equipped with NVIDIA Tesla K20m GPUs using the Ubuntu driver version 319.13 instead of the driver version 304.47.13 as used on the Cray XK7 TITAN. An update to CUDA 5.5 on TITAN is foreseen for October 2013 and we, therefore, had no other means of calculating floating point performance.

We measure the total execution time of one complete time step of the simulation by using the C function gettimeofday. With this information we compute the floating operations per second, FLOP/s, measuring the execution time for each total time step on the Cray XK7 TITAN. We were thus able to calculate the actual number of FLOP/s on TITAN although no GPU FLOP metrics were available on the Cray XK7 TITAN. Our communication system requires fixing the number of cells to a multiple of 8 in x-direction and y-direction and a multiple of 4 in z-direction. Moreover, we require a minimum of 24 x 24 x 12 cells per GPU. Thus, our maximum number of nodes and, accordingly, GPUs that we used was 18432. This also required adopting the number of cells in some of the scalings as detailed in the performance results section of this paper.

6. SYSTEM & ENVIRONMENT WHERE PERFORMANCE WAS MEASURED

Except for counting the floating point operations that required CUDA 5.5, all measurements were performed on the Cray XK7 TITAN system at Oak Ridge National Laboratory using CUDA 5.0 and the NVIDIA driver version 304.47.13. TITAN comprises of a total of 18688 nodes, each including a 16-core AMD Opteron 6274 CPU with 32 GB of 1600 MHz DDR3 ECC memory and an NVIDIA Tesla K20X Graphics Processing Unit with 6 GB GDDR5 ECC memory. The nodes are interconnected via the Cray GEMINI interconnect in a three-dimensional torus topology. PIConGPU was compiled using the Gnu C++ compiler version 4.6.3 and CRAY-MPICH 2 Version 5.6.3 running in the Cray Linux Environment. Our own system on which we counted the number of floating point operations performed in one time step of PIConGPU comprises of 17 nodes, each equipped with two 4-core Intel XEON E52606 with 64 GB of 1600 MHz DDR3 ECC memory and four NVIDIA Tesla K20m GPUs with 5 GB GDDR5 ECC memory, interconnected via Infiniband. We assume the number of floating point operations per time step to be fixed and only the execution time per time step to change when migrating from our system to TITAN.

7. PERFORMANCE RESULTS

7.1 Weak scaling and weak efficiency

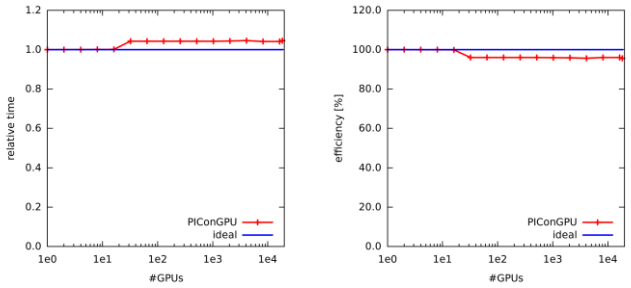

Figure 4. Weak scaling and efficiency when scaling PIConGPU from 1 to 18432 nodes.

The weak scaling was based on a 64 x 64 x 64 cells per GPU volume with 8.38 x 10^6 macro-particles (16 protons and 16 electrons per cell), using all 4.19 x 10^6 electrons to compute the radiation for 241 directions of observation and 1024 wavelengths. The number of directions and wavelengths was kept fixed when increasing the system size, while the number of particles and cells was increased when increasing the system size. The scaling starts with 1 GPU and ends up with a total number of 2048 x 2048 x 1024 cells on 16384 nodes, see Figure 4.

For the 18432 nodes data entry we extended the simulated volume to a total of 2048 x 2048 x 1152 cells. The only significant increase in execution time of 4% was observed when scaling from 16 to 32 nodes. This increase does not coincide with the number of nodes per blade (4) or the number of blades per cabinet (24) in TITAN. We thus attribute this to the GPU coverage of the simulation volume which changes from 4 x 2 x 2 GPUs on 16 nodes to 4 x 4 x 2 GPUs on 32 nodes. The weak efficiency when scaling from 1 to 16384 nodes was 96.06%, when scaling from 1 to 18432 nodes it was 95.61%.

7.2 Strong scaling and strong efficiency

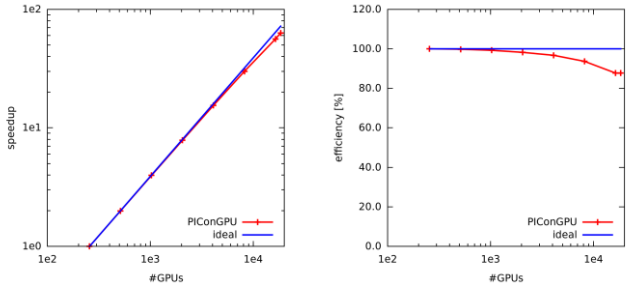

Figure 5. Strong scaling speed up and efficiency, setting A, scaling from 256 to 18432 nodes

For the strong scaling we used three slightly differing settings. **Setting A** uses 2048 x 2048 x 512 cells with 2 electrons and 2 protons per cell, giving a total number of 8.59 x 10^9 particles. Radiation was calculated for 241 directions and 1024 wavelengths. For 18432 nodes we had to extend the volume to 2048 x 2048 x 576 cells as a demand from our communication system. The results are shown in Figure 5. We divided the 16384 nodes into a 32 x 32 x 16 nodes volume, while for 18432 nodes we used a 32 x 32 x 18 nodes volume. For a scaling from 256 to 16384 nodes we measured a speedup of 56.14 (ideal: 64) and a strong efficiency of 87.71%, while scaling from 256 to 18432 nodes resulted in a speedup of 63.15 (ideal: 72) with 87.7 efficiency.

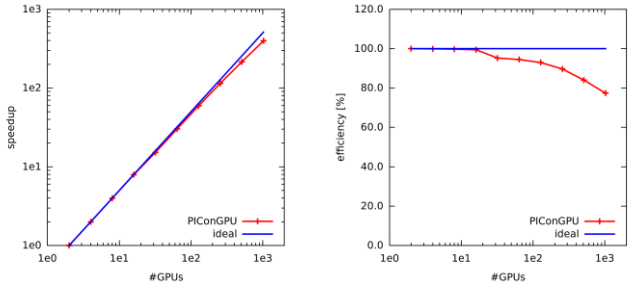

Figure 6. Strong scaling speed up and efficiency, setting B, scaling from 2 to 1024 nodes

Setting B used a 256 x 256 x 256 cells volume with 2 electrons and 2 protons per cell, scaling from 2 to 1024 nodes. The measured speedup was 396.02 (ideal: 512) and strong efficiency was 77.35%, as can be seen in Figure 6. **Setting C** used a 768 x 768 x 320 cells volume with 2 electrons and 2 protons per cell, scaling from 16 to 16384 nodes. Similar to setting A, we extended the simulation volume to 768 x 768 x 324 cells when scaling to 18432 nodes. Scaling from 16 to 16384 nodes we reached a speedup of 746.17 (ideal: 1024) and a strong efficiency of 72.87%, when scaling from 16 to 18432 nodes we measured a speedup of 794 (ideal: 1152) and a strong efficiency of 68.92%, see Figure 7.

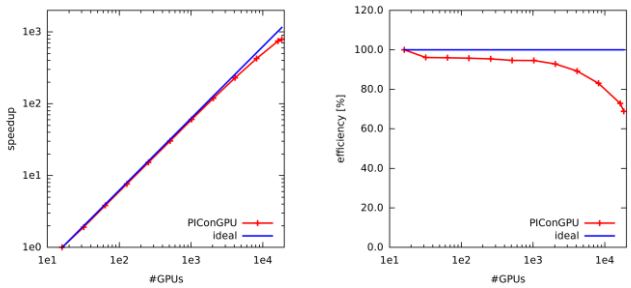

Figure 7. Strong scaling speed up and efficiency, setting C, scaling from 16 to 18432 nodes

7.3 Peak performance

PIConGPU is a fully template, mixed-precision code. We measure a combined peak floating point performance of 7.176 PFLOP/s (double-precision) plus 1.449 PFLOP/s (single-precision) based on a simulation volume of 8000 x 768 x 768 cells with 8 protons and 8 electrons per cell and half of all electrons used for radiation running on 18000 nodes. The overall percentage the simulation spent in kernels running on the GPU was 99.62% measured on 64 nodes using 256 x 256 x 256 cells, 8 protons and 8 electrons per cell and half of all electrons used for radiation.

7.4 Time to solution

With new results from production runs in July and August 2013 we have determined the duration for one time step by averaging over the time for steps 175 to 200 and 1020 to 1040, respectively. In the first case, the 25 time steps chosen took 425s, resulting in 17s per time step, while in the latter case 20 time steps took 372s, resulting in 18.6s per time step. Close to the end of the simulation one time step, averaged over step 1980 to 2000, took 30.7s.

This increase reflects the system's change from an ordered initial state into a fully unordered state driven by the KHI. These numbers do not include HDF5 output, which is usually totally asynchronous to computation. We have measured the duration of steps 980 to 1080 to be 2439s, including one HDF5 output, amounting to 24.39s per time step. Both the increase in time step duration including HDF5 output and the duration of a time step being longer for the production run than for the scaling runs stem from the fact that on TITAN no CUDA 5.5 drivers were available.

Older drivers have proven to be unstable with non-blocking operations. We decided for blocking execution of all kernels and deactivated asynchronous file output, increasing the stability of the simulation tremendously and allowing for long production runs without software failure. The total run time for 2000 time steps amounted to 16h. This includes 3h of I/O (checkpoints and restarts).

The file I/O strategy adopted for the production run will be replaced by fully asynchronous, parallel HDF5 file output and non-blocking kernel execution once CUDA 5.5 drivers are available.

8. FIRST RESULTS ON RADIATIVE SIGNATURES OF THE RELATIVISTIC KELVIN-HELMHOLTZ-INSTABILITY

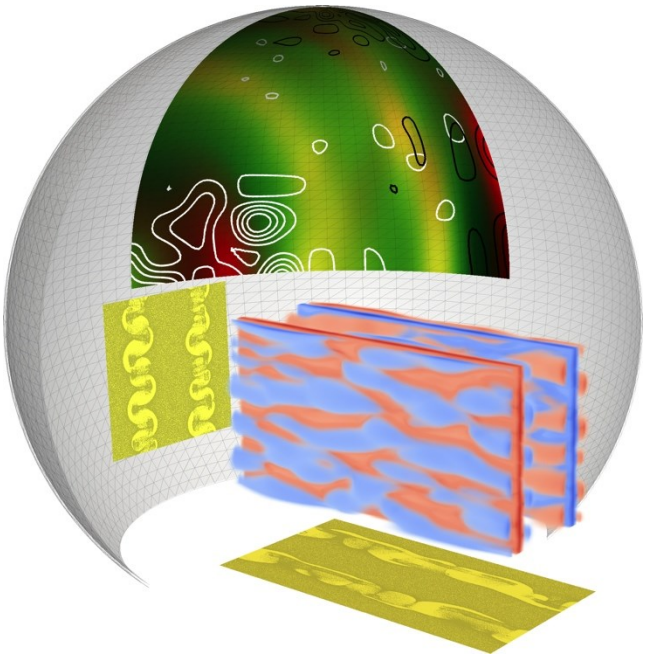

Figure 8. Visualization of our KHI simulation setup including radiation. Magnetic fields are given in blue and red while projections of the electron density are shown in yellow. The radiation emitted into one octant is projected onto a surrounding sphere.

Our setup uses periodic boundary conditions in all directions of the simulation volume. The inner plasma stream moves at a velocity corresponding to a relativistic γ-value of three in negative x-direction. The counter propagating outer stream is divided in two streams of half the width of the inner stream, one above the inner stream in y-direction, the other below. The z-axis lies in the plane of the interface between the streams and is normal to the plasma streams.

The KHI dynamics arise by an interplay between the electron motion and the generation of magnetic fields due to this motion, as these fields in turn drive the electrons via the v×B force, where v denotes the electron velocity and B the magnetic field strength.

A schematic of the simulation setup is depicted in Figure 8. The magnetic fields forming at the interface between the plasma streams are shown in blue and red. The electron density at the interfaces is shown in yellow in two projections. The radiation spectra are shown in projection onto a sphere surrounding the simulation volume.

As the KHI evolves, the energy spectrum of the electrons spreads out, see Figure 9. In the simulation we set the electron to ion mass ration to 1:1836, giving both particle species the same initial velocities. As the total energy in the simulation was conserved to a value better 99%, additional energy gained by the electrons mainly comes from the ions, consequently creating turbulence in the interface between the two plasmas due to local perturbations of the charge density.

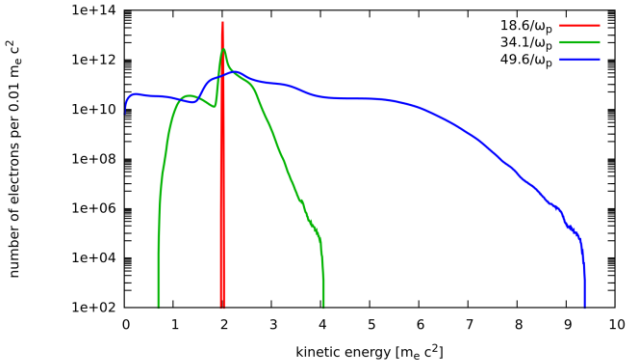

Figure 9. Electron energy spectrum at three different time steps as denoted in the legend. The red spectrum is close to the onset of the KHI growth, showing the small initial energy spread of electrons, while the green and blue energy spectra show the increasing energy spread over time.

We have set the initial conditions for the relativistic KHI such that the simulation volume contains a total of nine turbulent vortices forming at the interface between the two counter-propagating plasma flows due to shear forces [3, 25].

We reproduce these well-known KHI vortices developing in the electron density as seen in Figure 10 as well as the strong longitudinal magnetic fields forming along the shear surface depicted in Figure 12 similar to what has been observed in 3D MHD simulations [68].

The dominant longitudinal structure of these vortices is disturbed by an early onset of strong transverse magnetic dc-fields, see Figure 12, qualitatively confirming the results in [24]. These magnetic fields couple the longitudinal electron motion to the degree of motion transverse to the shear surface, driving the growth of a dominantly transverse instability as seen in Figure 11. These dynamics have transverse velocities of up to β_y=0.25 and are reflected in the far field radiation spectra computed during the evolution of the KHI.

In our simulation the magnetic dc-field saturates with a final strength of $3.5 m_e \omega_p / e$, a higher value than seen by [24], which might originate from electrons crossing the shear surface, as our phase space resolution is different to the one in [24], resulting in differing electron dynamics. This difference will be subject to further investigations beyond the scope of this publication. From the results presented here it is obvious that the transverse electron dynamics are equally important to the longitudinal dynamics. In the following we will consequently concentrate on the far field radiation emitted in both of these directions.

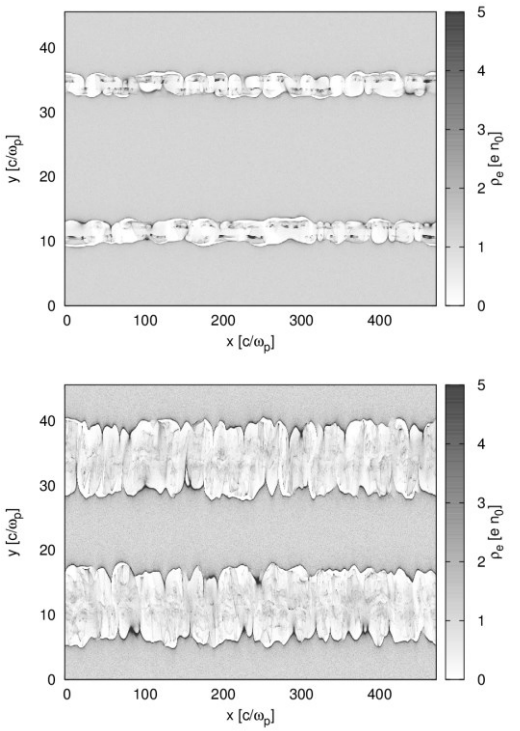

Figure 10. Electron density ρ_e in longitudinal direction along the shear surface at $t=34.1/\omega_p$ (upper figure) and $t=49.6/\omega_p$ (lower figure) in units of the initial electron density.

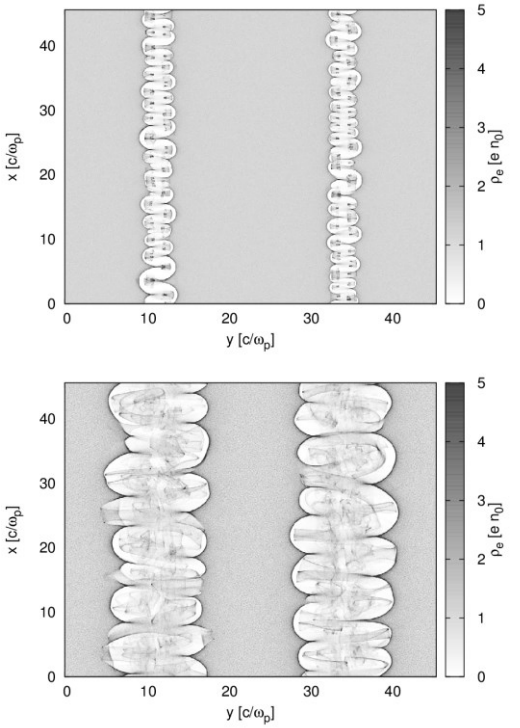

Figure 11. Electron density ρ_e in transverse direction perpendicular to the shear forces in units of the initial electron density at the same time steps as in Figure 10.

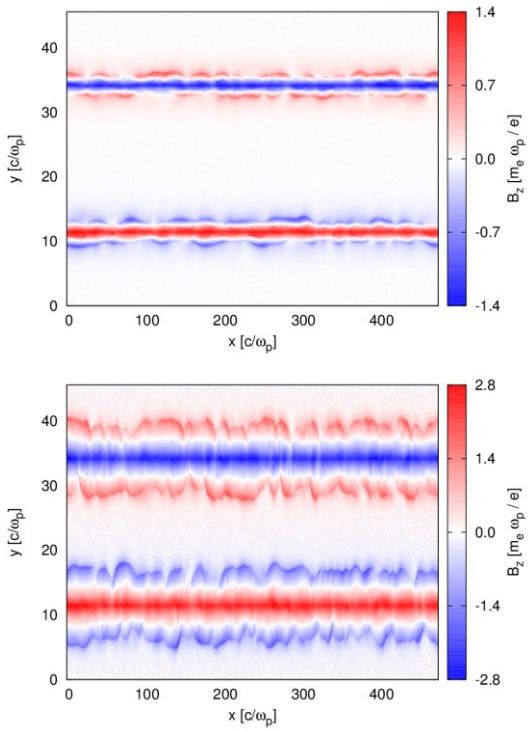

Figure 12. Strength of the longitudinal magnetic field in units of $m_e\omega_p/e$, corresponding to the same time steps as in Figure 10 and 11, $t=34.1/\omega_p$ (upper figure) and $t=49.6/\omega_p$ (lower figure), respectively.

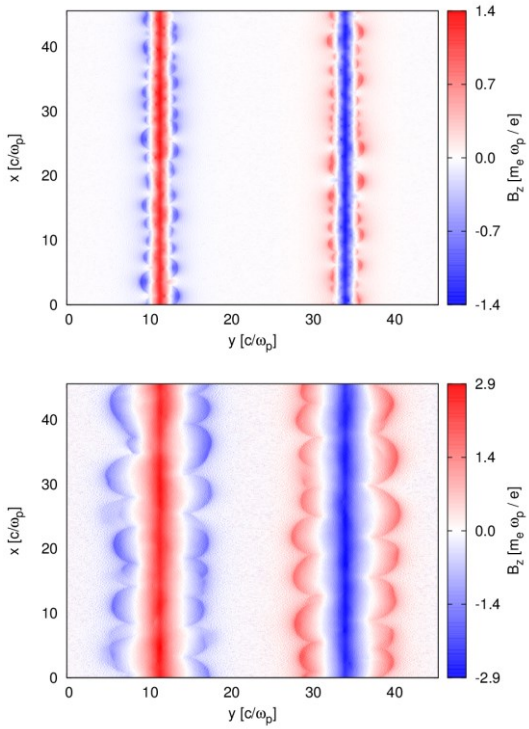

Figure 13. Strength of transverse magnetic dc-fields in units of $m_e\omega_p/e$, corresponding to the same time steps as in Figure 10 and 11 and 12.

8.1 Selected radiative signatures

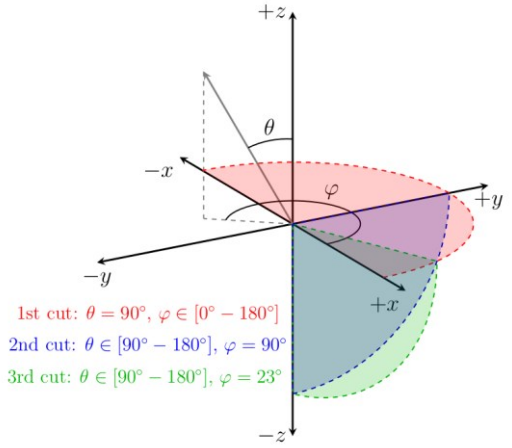

Figure 14. Choice of observation planes for the discussion of the far field radiation spectra.

The electromagnetic far field radiation shows clear spectral signatures that can be associated with the electron dynamics. In the following we discuss the temporal evolution of the spectra, concentrating on the radiation emitted into three different observation planes depicted in Figure 14, in the following referred to as 1st cut, 2nd cut and 3rd cut. The 1st cut (red) lies in the x-y plane and covers both radiation emitted along the flow direction of the plasma streams denoted by the x-axis and perpendicular to it in the direction of the y-axis. The 2nd cut (blue) is a plane normal to the plasma flows and covers radiation emitted perpendicular to the initial particle momenta. The 3rd cut (green) is similar to the 2nd, but also covers radiation emitted in the direction of the plasma flow. For the sake of simplicity, the spectral intensities have been given in SI-units for an initial electron density of $n_0=10^{25}$ m^{-3}.

During the buildup of the KHI, the electrons at the interface between the streams start to oscillate. This oscillation of relativistic electrons causes radiation in both the forward and transverse direction. Figure 15 shows the temporal evolution of the angular spectrum observed in the plane denoted by 1st cut in Figure 14. As this plane lies in the x-y plane and thus overlaps with the shear surface, we see a strong increase in radiation intensity during the evolution of the instability as the flow at the interface becomes more turbulent. This is expected as the shear forces act on the electrons, causing them to radiate.

We observe a strong peak in intensity in φ=90° direction perpendicular to the direction of plasma flow at $0.1\omega_p$, contrary to the naive expectation of strong radiation at the plasma frequency. This broad peak coincides with the frequency of the most unstable mode that is responsible for the formation of the nine vortices of the KHI. It is accompanied by higher harmonics indicating nonlinear motion of the electrons at the interface. At later times, the strong peak at $0.1\omega_p$ extends far into the plane orthogonal to the plasma flow, see Figure 16, correlating with an increasing transversal motion of the electrons. The closer the direction of emission of the radiation points into the direction of the plasma flow, the lower the frequency of the radiation is. This can be seen in Figure 17. Here, the emission into the plane denoted by "3rd cut" in Figure 14 is presented for time $t=34.1/\omega_p$, corresponding to the same time as the third graph from the top in Figure 15.

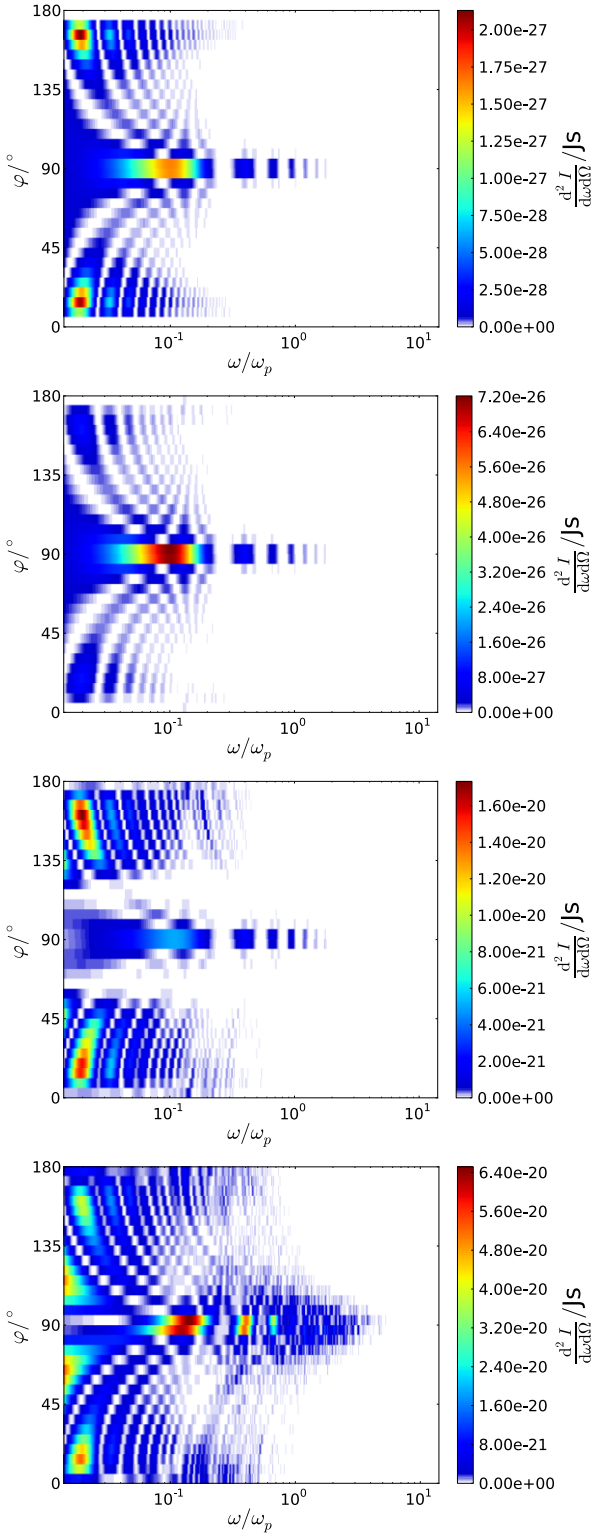

Figure 15. Radiation spectra for the observation plane denoted by "1st cut" at four consecutive time steps. From uppermost to lowermost figure: At $t=15.5/\omega_p$, $t=18.6/\omega_p$, $t=34.1/\omega_p$ and $t=49.6/\omega_p$. A clear peak at $0.1\omega_p$, φ=90° is visible at most times.

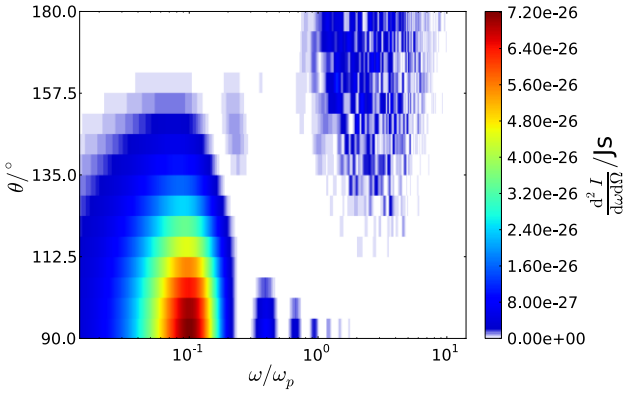

Figure 16. Spectrum corresponding to the observation plane denoted by „2nd cut" at $t=18.6/\omega_p$. The strong peak at $0.1\omega_p$ extends up to 40° into the plane orthogonal to the shear surface.

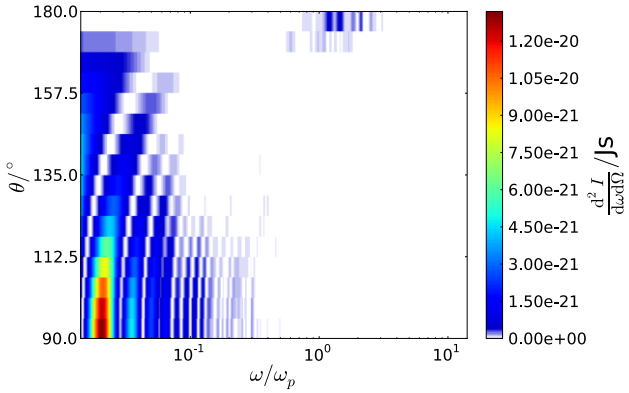

Figure 17. Spectrum of radiation emitted in the plane denoted by „3rd cut". With increasing angle θ the clearly visible intensity bands bend from lower to higher frequencies.

Clearly, one can see the radiation peak at $\varphi=23°$, $\theta=90°$ and $\omega\approx0.02\omega_p$ seen in Figure 15 at $t=34.1/\omega_p$ lying on a band bending towards higher frequencies for increasing θ, see Figure 17. We attribute this shift towards lower frequencies for $\theta=90°$ and $\varphi=0°$ or $\varphi=180°$ to the plasma flow, as it is strongest when the direction of emission of the radiation and the direction of the plasma flow coincide.

As can be seen from this first preliminary study, the dynamics of the electrons in the development of the KHI can be connected to the signatures in the radiation spectra. An analysis of the rich features of both the spectra and the turbulent electron motion, is well beyond the scope of this text and will be addressed in a subsequent publication.

9. IMPLICATIONS FOR FUTURE SYSTEMS AND APPLICATIONS

Plasma physics can greatly profit from the developments presented in this work. We show that it is possible to use the radiation emitted by all particles in a plasma as a new diagnostic to study the inner dynamics of the plasma. With this it is possible to give quantitative results on the spectral intensities observed, as they include coherence effects affecting the radiation pattern when electromagnetic waves from two regions of the plasma superimpose. This enables us to use relativistic Lorentz transformations to obtain the spectra for jets that have a highly relativistic velocity relative to the observer.

Spectra of such detail can only be obtained when calculated during the simulation, as storing the trajectory data needed to compute the spectra can easily amount to tens to hundreds of Petabyte, while limiting the performance of the simulation by the bandwidth of the file system. Our results prove that the complex particle-in-cell algorithm algorithms can be efficiently vectorized and implemented on accelerator hardware such as graphic processing units.

We can well hide the latency of all communication, requiring a minimum of one thread on the host CPU for inter-node communication. This means that 15 of the 16 cores in each TITAN node are idle. Utilizing the additional 10% of performance provided by CPUs would require asymmetric and adaptive communication and load balancing schemes.

Furthermore, the straightforward encapsulation of the physics model in CUDA kernels would have to be extended by supplying the same algorithms as a multi-threaded host function, thus resulting in unnecessary code redundancy. Hence, it was avoided at this time. As we only require the CPU for communication and file system access, the CPUs can potentially be replaced by processors with less power consumption and reduced capabilities, such as ARM processors [58, 1]. Communication via remote direct memory access (RDMA) in principle can allow for direct GPU to GPU communication [49], even across nodes via network using MPI [65]. From these developments one can imagine to have high performance computing systems consisting solely of GPUs or other accelerator hardware. Codes such as PIConGPU would strongly benefit from this development as they are fully vectorized and make optimum use of the large bandwidth and floating point performance that vector computations can provide.

Although PIConGPU is an implementation of the particle-in-cell algorithm used for relativistic plasma simulations, the techniques presented in the section on innovations can be directly adopted for optimizing general particle-mesh codes [56] for many-core systems. As all of the techniques presented in this paper can be described by abstract concepts, consequent use of compile-time polymorphism and meta-programming techniques helps to isolate those parts of the code that have to be adapted to a specific hardware for optimizing performance. Demanding data locality ensures that communication can be abstracted and all parts of the simulation only have to touch the data in a single super cell. As an example of hardware-dependent optimization, one could use the size of the super cell as a parameter that is adjusted to the hardware-specific size of a vector.

Moreover, vectorized data structures such as the super cell and the attribute tile can be utilized to optimize load balancing when using task-parallel execution. For future systems, resilience could be achieved by computing subsets of the overall simulation volume more than once, distributing the vectorized data structure to the nodes. For Exascale compute systems communication latency potentially increases and thus it could be favourable to compute rather than communicate, thereby duplicating results. Copies of the data residing on different nodes could then be used if any of the nodes fail. This however requires the number of nodes that communicate with each other to change dynamically.

10. ACKNOWLEDGEMENTS

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.
https://www.olcf.ornl.gov/kb_articles/olcf-acknowledgement/

11. REFERENCES

[1] B. S. C. Alex Ramirez. Pedraforca: A first arm + gpu cluster for hpc. http://nvidia.fullviewmedia.com/gtc2013/0319-210E-S3064.html, 2013.

[2] A. Alexandrescu. *Modern C++ Design: Generic Programming and Design Patterns Applied*. Addison-Wesley Professional, 2001.

[3] E. P. Alves, T. Grismayer, S. F. Martins, F. Fiuza, R. A. Fonseca, and L. O. Silva. Large-scale magnetic field generation via the kinetic kelvin-helmholtz instability in unmagnetized scenarios. *The Astrophysical Journal Letters*, 746(2):L14, 2012.

[4] S. Bastrakov, R. Donchenko, A. Gonoskov, E. Efimenko, A. Malyshev, I. Meyerov, and I. Surmin. Particle-in-cell plasma simulation on heterogeneous cluster systems. *Journal of Computational Science*, 3(6):474–479, 2012.

[5] J. Boris. Relativistic plasma simulation-optimization of a hybrid code. In *Proceedings of the 4th Conference on Numerical Simulation of Plasmas. Naval Res. Lab., Washington, D.C.*., pages 3–67, 1970.

[6] K. Bowers, B. Albright, B. Bergen, L. Yin, K. Barker, and D. Kerbyson. 0.374 pflop/s trillion-particle kinetic modeling of laser plasma interaction on roadrunner. In *High Performance Computing, Networking, Storage and Analysis, 2008. SC 2008. International Conference for*, pages 1–11, 2008.

[7] H. Burau, R. Widera, W. HoÌ‚nig, G. Juckeland, A. Debus, T. Kluge, U. Schramm, T. Cowan, R. Sauerbrey, and M. Bussmann. Picongpu: A fully relativistic particle-in-cell code for a gpu cluster. *Plasma Science, IEEE Transactions on*, 38(10):2831–2839, 2010.

[8] M. Bussmann. Picongpu. http://picongpu.hzdr.de, 2013.

[9] I. Chapman, S. Brown, R. Kemp, and N. Walkden. Toroidal velocity shear kelvin-helmholtz instabilities in strongly rotating tokamak plasmas. *Nuclear Fusion*, 52(4):042005, 2012.

[10] G. Chen, L. ChacÃ³n, and D. Barnes. An efficient mixed-precision, hybrid cpu-gpu implementation of a nonlinearly implicit one-dimensional particle-in-cell algorithm. *Journal of Computational Physics*, 231(16):5374 – 5388, 2012.

[11] M. Chen, E. Esarey, C. G. R. Geddes, C. B. Schroeder, G. R. Plateau, S. S. Bulanov, S. Rykovanov, and W. P. Leemans. Modeling classical and quantum radiation from laser-plasma accelerators. *Phys. Rev. ST Accel. Beams*, 16:030701, 2013.

[12] J. M. Dawson. Particle simulation of plasmas. *Rev. Mod. Phys.*, 55:403–447, Apr 1983.

[13] P. A. Delamere, R. J. Wilson, S. Eriksson, and F. Bagenal. Magnetic signatures of kelvin-helmholtz vortices on saturn's magnetopause: Global survey. *Journal of Geophysical Research: Space Physics*, 118(1):393, 2013.

[14] J. Eastwood, W. Arter, N. Brealey, and R. Hockney. Body-fitted electromagnetic pic software for use on parallel computers. *Computer Physics Communications*, 87(1-2):155 – 178, 1995.

[15] J. W. Eastwood. The virtual particle electromagnetic particle-mesh method. *Computer Physics Communications*, 64(2):252 – 266, 1991.

[16] T. Esirkepov. Exact charge conservation scheme for particle-in-cell simulation with an arbitrary form-factor. *Computer Physics Communications*, 135(2):144 – 153, 2001.

[17] R. L. Fermo, J. F. Drake, and M. Swisdak. Secondary magnetic islands generated by the kelvin-helmholtz instability in a reconnecting current sheet. *Phys. Rev. Lett.*, 108:255005, 2012.

[18] F. Fiuza and et al. Record simulations conducted on lawrence livermore supercomputer. https://www.llnl.gov/news/-newsreleases/2013/Mar/NR-13-03-05.html, 2013.

[19] C. Foullon, E. Verwichte, K. Nykyri, M. J. Aschwanden, and I. G. Hannah. Kelvin-helmholtz instability of the cme reconnection outflow layer in the low corona. *The Astrophysical Journal*, 767(2):170, 2013.

[20] J. T. Frederiksen, T. Haugbolle, M. V. Medvedev, and A. Nordlund. Radiation spectral synthesis of relativistic filamentation. *The Astrophysical Journal Letters*, 722(1):L114, 2010.

[21] G.-T. GmbH. Vampir and vampir trace. http://-www.vampir.eu/, 2013.

[22] L. E. H. Godfrey, J. E. J. Lovell, S. Burke-Spolaor, R. Ekers, G. V. Bicknell, M. Birkinshaw, D. M. Worrall, D. L. Jauncey, D. A. Schwartz, H. L. Marshall, J. Gelbord, E. S. Perlman, and M. Georganopoulos. Periodic structure in the megaparsec-scale jet of pks 0637-752. *The Astrophysical Journal Letters*, 758(2):L27, 2012.

[23] A. D. Greenwood, K. L. Cartwright, J. W. Luginsland, and E. A. Baca. On the elimination of numerical cerenkov radiation in pic simulations. *Journal of Computational Physics*, 201(2):665 – 684, 2004.

[24] T. Grismayer, E. P. Alves, R. A. Fonseca, and L. O. Silva. dc-magnetic-field generation in unmagnetized shear flows. *Phys. Rev. Lett.*, 111:015005, Jul 2013.

[25] A. Gruzinov. GRB: magnetic fields, cosmic rays, and emission from first principles? *ArXiv e-prints*, Mar. 2008.

[26] T. Haugboelle, J. T. Fredriksen, and A. Nordlund. Photon-Plasma: a modern high-order particle-in-cell code. *arXiv*, astro-ph.SR:1211.4575, 2012.

[27] HDF-Group. Hdf5. http://www.hdfgroup.org/HDF5/, 2013.

[28] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles*. Taylor & Francis, 1989.

[29] W. Hönig, F. Schmitt, R. Widera, H. Burau, G. Juckeland, M. S. Müller, and M. Bussmann. A generic approach for developing highly scalable particle-mesh codes for gpus. In *Symposium on Application Accelerators in High-Performance Computing 2012, SAAHC'12, Argonne National Laboratory*, pages 1–3, 2012.

[30] O. A. Hurricane, V. A. Smalyuk, K. Raman, O. Schilling, J. F. Hansen, G. Langstaff, D. Martinez, H.-S. Park, B. A. Remington, H. F. Robey, J. A. Greenough, R. Wallace, C. A. Di Stefano, R. P. Drake, D. Marion, C. M. Krauland, and C. C. Kuranz. Validation of a turbulent kelvin-helmholtz shear layer model using a high-energy-density omega laser experiment. *Phys. Rev. Lett.*, 109:155004, 2012.

[31] J. D. Jackson. *Classical Electrodynamics*. John Wiley & Sons, Inc., 1999.

[32] R. Joseph, G. Ravunnikutty, S. Ranka, E. D'Azevedo, and S. Klasky. Efficient gpu implementation for particle in cell algorithm. In *Parallel Distributed Processing Symposium (IPDPS), 2011 IEEE International*, pages 395–406, 2011.

[33] X. Kong, M. C. Huang, C. Ren, and V. K. Decyk. Particle-in-cell simulations with charge-conserving current deposition on graphic processing units. *Journal of Computational Physics*, 230(4):1676–1685, 2011.

[34] Y. Kuramitsu, Y. Sakawa, S. Dono, C. D. Gregory, S. A. Pikuz, B. Loupias, M. Koenig, J. N. Waugh, N. Woolsey, T. Morita, T. Moritaka, T. Sano, Y. Matsumoto, A. Mizuta,

N. Ohnishi, and H. Takabe. Kelvin-helmholtz turbulence associated with collisionless shocks in laser produced plasmas. *Phys. Rev. Lett.*, 108:195004, 2012.

[35] G. Lapenta, V. Pierrard, R. Keppens, S. Markidis, S. Poedts, O. Šebek, P. M. Trávncek, P. Henri, F. Califano, F. Pegoraro, M. Faganello, V. Olshevsky, A. L. Restante, Å. Nordlund, J. Trier Frederiksen, D. H. Mackay, C. E. Parnell, A. Bemporad, R. Šusino, and K. Borremans. SWIFF: Space weather integrated forecasting framework. *Journal of Space Weather and Space Climate*, 3(26):A260000, 2013.

[36] E. Liang, M. Boettcher, and I. Smith. Magnetic field generation and particle energization at relativistic shear boundaries in collisionless electron-positron plasmas. *The Astrophysical Journal Letters*, 766(2):L19, 2013.

[37] K. Madduri, E.-J. Im, K. Z. Ibrahim, S. Williams, S. Ethier, and L. Oliker. Gyrokinetic particle-in-cell optimization on emerging multi- and manycore platforms. *Parallel Computing*, 37(9):501 – 520, 2011.

[38] P. Mertmann, D. Eremin, T. Mussenbrock, R. P. Brinkmann, and P. Awakowicz. Fine-sorting one-dimensional particle-in-cell algorithm with monte-carlo collisions on a graphics processing unit. *Computer Physics Communications*, 182(10):2161 – 2167, 2011.

[39] J. Moen, K. Oksavik, L. Alfonsi, Y. Daabakk, V. Romano, and L. Spogli. Space weather challenges of the polar cap ionosphere. *Journal of Space Weather and Space Climate*, 3(26):A260000, 2013.

[40] U. V. Möstl, M. Temmer, and A. M. Veronig. The kelvin-helmholtz instability at coronal mass ejection boundaries in the solar corona: Observations and 2.5d mhd simulations. *The Astrophysical Journal Letters*, 766(1):L12, 2013.

[41] MPI-Forum. Message passing interface standard. http://-www.mpi-forum.org/, 2013.

[42] K. I. Nishikawa, P. Hardee, B. Zhang, I. Dutan, M. Medvedev, et al. Radiation from accelerated particles in relativistic jets with shocks, shear-flow, and reconnection. *arXiv*, astro-ph.HE:1303.2569, 2013.

[43] K.-I. Nishikawa, J. Niemiec, M. Medvedev, B. Zhang, P. Hardee, Y. Mizuno, A. Nordlund, J. Frederiksen, H. Sol, M. Pohl, D. H. Hartmann, and G. J. Fishman. Simulation of relativistic shocks and associated radiation from turbulent magnetic fields. *Proceedings of the International Astronomical Union*, 6:354–357, 8 2010.

[44] NVIDIA. Nvidia command line profiler. https://-developer.nvidia.com/nvidia-visual-profiler, 2013.

[45] NVIDIA. Nvidia cuda. https://developer.nvidia.com/-what-cuda, 2013.

[46] NVIDIA. Nvidia cuda 5.5 early access. https://-nvdeveloper.nvidia.com, 2013.

[47] NVIDIA. Nvidia cuda profiling tools interface. https://-developer.nvidia.com/cuda-profiling-tools-interface, 2013.

[48] NVIDIA. Nvidia cuda programming guide. http://-docs.nvidia.com/cuda/cuda-c-programming-guide/index.html, 2013.

[49] NVIDIA. Rdma for gpu direct. http://docs.nvidia.com/-cuda/gpudirect-rdma/index.html, 2013.

[50] ORNL. Titan. http://www.olcf.ornl.gov/titan/, 2013.

[51] J. Paral and R. Rankin. Dawn-dusk asymmetry in the kelvin-helmholtz instability at mercury. *Nature Communications*, 4:1645, 2013.

[52] R. Pausch. Electromagnetic Radiation from Relativistic Electrons as Characteristic Signature of their Dynamics. Master's thesis, Technical University Dresden, Dresden, Germany, 2012.

[53] P. Ricci and B. N. Rogers. Plasma turbulence in the scrape-off layer of tokamak devices. *Physics of Plasmas*, 20(1):010702, 2013.

[54] E. Roediger, R. P. Kraft, M. E. Machacek, W. R. Forman, P. E. J. Nulsen, C. Jones, and S. S. Murray. Irregular sloshing cold fronts in the nearby merging groups ngc 7618 and ugc 12491: Evidence for kelvin-helmholtz instabilities. *The Astrophysical Journal*, 754(2):147, 2012.

[55] E. Rutter, M. Grosskopf, G. Malamud, C. Kuranz, E. Harding, P. Keiter, and R. Drake. Comparison between kelvin-helmholtz instability experiments on omega and simulation results using the crash code. *High Energy Density Physics*, 9(1):148, 2013.

[56] I. F. Sbalzarini, J. H. Walther, B. Polasek, P. Chatelain, M. Bergdorf, S. E. Hieber, E. M. Kotsalis, and P. Koumoutsakos. A software framework for the portable parallelization of particle-mesh simulations. In *in Proceedings of Euro-Par 2006, LNCS 4128, 730-739*, 2006.

[57] F. K. Schinzel, A. P. Lobanov, and J. A. Zensus. The long-term evolution of the parsec scale jet of the quasar 3c345. *Journal of Physics: Conference Series*, 372(1):012070, 2012.

[58] N. . SECO. Cuda development kit for gpu-accelerated arm-based systems. http://www.nvidia.com/object/seco-dev-kit.html, 2013.

[59] L. Sironi and A. Spitkovsky. Synthetic spectra from particle-in-cell simulations of relativistic collisionless shocks. *The Astrophysical Journal Letters*, 707(1):L92, 2009.

[60] G. Stantchev, W. Dorland, and N. Gumerov. Fast parallel Particle-To-Grid interpolation for plasma PIC simulations on the GPU. *Journal of Parallel and Distributed Computing*, 68(10):1339–1349, 2008.

[61] W. Thomson. Hydrokinetic solutions and observations. *Philosophical Magazine*, 42:362, 1871.

[62] H. v. Helmholtz. Über discontinuierliche flüssigkeits-bewegungen. *Monatsberichte der Königlichen Preussische Akademie der Wissenschaften zu Berlin*, 23:215, 1868.

[63] J.-L. Vay. Simulation of beams or plasmas crossing at relativistic velocity. *Physics of Plasmas*, 15(5):056701, 2008.

[64] J. Villasenor and O. Buneman. Rigorous charge conservation for local electromagnetic field solvers. *Computer Physics Communications*, 69(2-3):306–316, 1992.

[65] H. Wang, S. Potluri, M. Luo, A. K. Singh, S. Sur, and D. K. Panda. Mvapich2-gpu: optimized gpu to gpu communication for infiniband clusters. *Comput. Sci.*, 26(3-4):257–266, 2011.

[66] A. Woods and L. Ludeking. Performance enhancement of magic fdtd-pic plasma-wave simulations using gpu processing. In *Plasma Science, 2010 Abstracts IEEE International Conference on*, pages 1–1, 2010.

[67] K. Yee. Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media. *Antennas and Propagation, IEEE Transactions on*, 14(3):302–307, 1966.

[68] W. Zhang, A. MacFadyen, and P. Wang. Three-dimensional relativistic magnetohydrodynamic simulations of the kelvin-helmholtz instability: Magnetic field amplification by a turbulent dynamo. *The Astrophysical Journal Letters*, 692(1):L40, 2009.